AMENDMENTS TO THE CLAIMS

1-44. (Cancelled)

45. (Currently Amended) A method of encrypting multi-media data flow packets, comprising the steps of:

receiving a series of multi-media data flow packets, each packet comprising a sequence number;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

46. (Previously Presented) The method of claim 45, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

47. (Previously Presented) The method of claim 45, further comprising the step of performing bit manipulation within said first multi-media data flow packet.

48. (Previously Presented) The method of claim 47, wherein said step of performing bit manipulation is performed by using a bit-size operation that is restorable.

49. (Previously Presented) The method of claim 48, wherein said bit-size operation comprises negation.

50. (Previously Presented) The method of claim 45, further comprising the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

51. (Previously Presented) The method of claim 50, wherein said destination address is a destination port address of said second endpoint.

52. (Currently Amended) A computer readable medium for encrypting multi-media data flow packets, the program ~~comprising logic~~ for performing the steps of:

receiving a series of multi-media data flow packets;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

53. (Previously Presented) The computer readable medium of claim 52, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

54. (Previously Presented) The computer readable medium of claim 52, the program further comprising logic for performing the step of performing bit manipulation within said first multi-media data flow packet.

55. (Previously Presented) The computer readable medium of claim 54, wherein said step of performing bit manipulation is performed by using a bit-size operation that is restorable.

56. (Previously Presented) The computer readable medium of claim 55, wherein said

bit-size operation comprises negation.

57. (Previously Presented) The computer readable medium of claim 52, the program further comprising logic for performing the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

58. (Previously Presented) The computer readable medium of claim 57, wherein said destination address is a destination port address of said second endpoint.

59. (Currently Amended) A system for encrypting multi-media data flow packets, comprising:

a transceiver;

software stored within said first endpoint defining functions to be performed by the system; and

a processor configured by said software to perform the steps of:

receiving a series of multi-media data flow packets;

storing the series of multi-media data flow packets in a jitter buffer;

re-sequencing the series of multi-media data flow packets into a pseudo-random order; and

transmitting each multi-media data flow packet in the re-sequenced series in the re-sequenced order.

60. (Currently Amended) The system of claim [[45]] 59, wherein said re-sequencing uses a randomization code that is algorithmically predictable if a key to said randomization code is known.

61. (Previously Presented) The system of claim 59, processor configured by said software to perform the step of pseudo-randomly shuffling a destination address of each of the multi-media data flow packets.

62. (Previously Presented) The system of claim 61, wherein said destination address is a destination port address of said second endpoint.

63. (Previously Presented) A method of encrypting a multi-media data flow packet, comprising the steps of:

storing a first multi-media data flow packet, the packet comprising bytes in a first order;

generating a non-duplicating pseudo-random sequence of integers, the sequence containing M integers, each integer between 1 and M;

reordering at least a portion of the bytes of the first packet into a new order specified by the integers in the generated sequence; and

transmitting the reordered multi-media data flow packet.

64. (Previously Presented) The method of claim 63, wherein M is equal to the maximum size of the first multi-media data flow packet.

65. (Previously Presented) The method of claim 63, wherein M is less than the maximum size of the first multi-media data flow packet.

66. (Previously Presented) The method of claim 63, each byte in the first multi-media data packet associated with an index, wherein the reordering step comprises the steps of:

copying the byte associated with the current index position of the packet into a new index position within the packet, the new index position in the packet equal to the integer at the current index position within the generated sequence;

updating the current index position to the next index position; and

repeating the copying step until the portion has been reordered.

67. (Currently Amended) A method of encrypting a series of multi-media data flow packets, comprising the steps of:

receiving a series of multi-media data flow packets belonging to a first flow, each packet in the series having the same port address;

generating a pseudo-random sequence of numbers, the sequence associated with the port address;

replacing the port address in each packet with ~~a value which is a function of~~ the corresponding number in the sequence or the product of the corresponding number in the sequence and the size of the sequence; and

transmitting each packet to a receiver.

68. (Cancelled)

69. (Cancelled)

70. (Previously Presented) The method of claim 67, wherein the generating step uses a randomization code that is predictable if a key to the randomization code is known.

71. (Previously Presented) The method of claim 70, wherein the key is known to the receiver.

72. (Previously Presented) The method of claim 67, wherein the size of the sequence is known to the receiver.

73. (Previously Presented) The method of claim 67, wherein the port address comprises a destination port address.